

ALGORITHM IN LATTICE C TO GENERATE S(n)

by Jim Duncan
9 Ryeground Lane
Formby
Liverpool
L37 7EG ENGLAND

Computer: Atari 1040ST
Run time: ca. 50 hrs to generate S(n) for 1 000 000 numbers

```
#include <stdio.h>
#include <math.h>

unsigned long int pst,ast,s,t;

main()
{
    long int n;
    FILE *fp;

    printf("input n (1 <= n < 2 147 483 647)\n");
    scanf("%ld", &n);
    fp = fopen("PRN:", "w");
    fprintf(fp, "n = %ld  S(n) = %ld\n", n, smaran(n));
    fclose(fp);

}

smaran(m)
unsigned long int m;
{
    unsigned long int mst,p,a,fact;
    double r;

    if (m == 1)
        return(0);
    else { /* STANDARD FORM of m */
        r = mst = m;
        p = 1;
        fact = 1;
        while ( ++p <= sqrt(r)) {
            a = 0;
            while (mst % p == 0) {
                mst = mst/p;
                a++;
            }
            r = mst;
            if (a > 0) {
                pst = p;
                ast = a;
                t = s = 0;
            }
        }
    }
}
```

```

        tors3(); /* find smallest factorial (t) with */
        /* p^a divisor */
        if (t > fact)
            fact = t;
        }
    }
    if (mst > fact)
        fact = mst;
    return(fact);
}
tors1()
{
    unsigned long int i;

/* test number is pst^ast */
/* s is the difference between a factorial number t and ast*(pst-1) */
/* s forms a pattern which determines the smallest value of t for which */
/* the test number is a divisor */

    i = 0;
    while (++i < pst*pst && t-s < ast*(pst-1)) {
        if (i % pst == 0)
            s = s-pst+2;
        else
            s++;
        t += pst;
    }
}

tors2()
{
    unsigned long int i;

    tors1();
    i = 0;
    while (++i < pst*pst && t-s < ast*(pst-1)) {
        if (i % pst == 0)
            s = s-3*pst+4;
        else
            s = s-2*pst+3;
        t += pst;
        tors1();
    }
}

tors3()
{
    unsigned long int i;

    tors2();
    i = 0;
    while (++i < pst*pst && t-s < ast*(pst-1)) {
        if (i % pst == 0)
            s = s-5*pst+6;
        else
            s = s-4*pst+5;
        t += pst;
        tors2();
    }
}

```